Brain Tumor Image Classification using transfer learning.
UC3MAL101 Final Assessment

Martin Slaastuen

ABSTRACT

Background: CNS tumors are one of the leading causes of tumor deaths in the age group 0-19 and brain tumors often require invasive procedures to grade a tumor. Using Deep Learning to create a method of classifying tumors reduces risk for patients and can assist radiologist with diagnosis.
Method: Dataset that contains 7022 MRI images with four different labels, glioma, meningioma, no tumor and pituitary tumors are used to train and test three different CNN architectures, EfficientNet B0, VGG16 and ResNet50. The data is split into a train set for training the model and a test set to evaluate the model.
Results: Evaluation of the three CNN models got a result of 98% accuracy for the EfficientNet model, 92% for the VGG model and 80% for the ResNet50 model.
Conclusion: After comparing the classification results the EfficientNet B0 model show a promising result for classifying brain tumors.

1 INTRODUCTION

1.1 Problem Statement

In recent years deep learning and especially Convolutional Neural Networks (CNN) have been used for visual recognition tasks like analysing and classifying brain tumors for example [2, 8, 12, 13]. Brain tumor grading is usually done by performing invasive procedures like biopsy, so by having a non-invasive procedures like classify brain tumors is preferred and poses less risk for the patients[13]. Tumors in the central nervous system are the leading cause of tumor mortality in the age group 0-19 and second leading cause for the age group 20-39[1]. Being able to classify brain tumors using machine learning and deep learning techniques like CNN to help radiologist may give more accurate diagnosis of brain tumors and can also reduce the workload of radiologist and reduce misdiagnosis caused by overworked or tired radiologist[2]. The purpose of this paper is to create a machine learning model to classify brain tumors. The author wants to compare the three model architectures VGG16, ResNet50 and EfficientNetB0 to see if EfficientNet can produce an accurate classifier for classifying brain tumors by using transfer learning approach.

1.2 Related Work

Convolutional Neural Network was first proposed to use for handwritten digits recognition in 1990 by Yann LeCun[14], so CNN is not a new concept. Many studies have done research on CNN architectures like VGG and ResNet for classifying brain tumors with good results, for example[8, 12, 13]. [Gopal S. Tandel, Ashish Tiwari, and O. G. Kakde. 2021] reached an accuracy of 95% on both VGG16 and ResNet50 with their proposed framework[13]. While [Venkatesan Rajinikanth, Seifedine Kadry, and Yunyoung Nam. 2021] reached a very high accuracy with VGG16 and their framework of over 98% which is extremely good[12]. The author found found an interesting study for classifying Chest X-rays for diagnosing COVID-19 patient using a different architecture called EfficientNet [9]. Reaching accuracy of 99,62% for binary classification and 96,69% for multiclassification. However the author found very little research done on EfficientNet architecture for classifying brain tumors. EfficientNet has 8 models from B0 to B7. The difference between the models are different amount of parameters ranging from 5,3 million to 66 million[9].

Training CNN require a large amount of data Studies show that by using transfer learning, which is an approach where the learning process of a new problem is increased via prior knowledge that is already gained by a model, CNN using transfer learning outperforms CNNs that has been trained from scratch[13].

2 DATASET

2.1 Dataset Information

The dataset contains 2D MRI images of different kinds of brain tumors. It has 7022 MRI images which is classified into four different classes, Glioma tumors, Meningioma tumors, Pituitary tumors and No tumors. The images are split into two folders, one for training and one for testing. Looking at the information about the dataset[10], it is stated that 22% of the images are in the testing folder and is intended to use for testing. For this given task the dataset also contains a predict folder with 40 images that will be used to predict what kind of tumor is in the images using one of the CNN networks we have looked at in this paper. The dataset is licensed as public domain, and was created by Masoud Nickparvar[10].

2.2 Data Prepossessing

In the information about the dataset it is stated that the images are of different sizes, so we have to resize the images to the optimal size for our model. According to Keras documentation[6, 7] and an article from Yixing Fu from the keras team[3], all three models that was tested in this paper (VGG16, ResNet50, EfficientNetB0) has a default input shape of 224, 244 and 3 channels[3, 6, 7]. However, EfficientNet B0 and ResNet50 can accept other resolutions [3, 6]. Table 1 shows what size the images has been resized to for each model and Figure 1 show a sample image of each class in the dataset.

Model	Image Size
EfficientNetB0	224x224x3
VGG16	224x224x3
ResNet50	224x224x3

Table 1: Table that shows the image size used as input on each model.

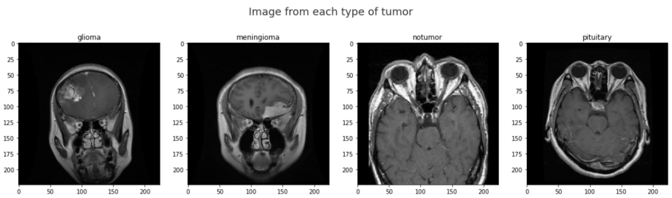

Figure 1: Sample of each brain tumor class in the dataset including no tumors.

3 METHODOLOGY

EfficientNet B0 has been chosen because it has less amount of parameters than the other EfficientNets and is suitable to the authors available resources. The other models VGG16 and ResNet50 has been chosen as they have been used in multiple studies with good results for example[2, 8, 12, 13]. Figure 2 show a simple block diagram of how the models should work.

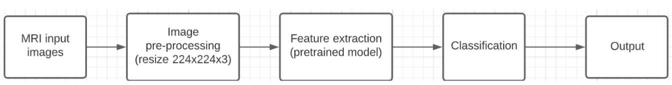

Figure 2: Simple Block Diagram of how the models should work.

The three models used in this paper has a few parameters that require tuning so we get the best model for our problem. In this paper all models are pretrained transfer learning models that uses weights from ImageNet. This means that the models has been trained on the ImageNet database. The ImageNet database contains 1000 classes and was created to provide a large scale image database for training and testing[4]. Transfer Learning is used to increase the performance of of a Deep Learning model and speed up the training, it does this by using prior knowledge that the model has already gained[13]. Since we use pretrained models we can tune the parameters in the layers added to to the models. The layers added for each model are listed below.

- **EfficientNet B0**
  - Global_average_Pooling_2D layer is used to reduce the number of parameters and by doing this we minimize overfitting [9].
  - Dropout Layer with a dropout rate of 0.5 has been chosen. This is to reduce feature dimension and to reduce the risk of overfitting[12].
  - Dense layer is added as an output layer with 4 neurons. We have four neurons because we have four classes to classify. This layer uses activation SoftMax as it is used for multi-class.
- **VGG16**
  - Flatten layer is used to flatten the output of the VGG16 model.
  - Dense Layer is added. GridSearchCV is used to to find optimal number of neurons in the layer.
  - Dropout Layer with a dropout rate of 0.5.
  - Dense Layer as output layer with 4 neurons.
- **ResNet50**
  - Flatten layer is used to flatten the output of the ResNet50 model.
  - Dense Layer is added with ReLu activation. GridSearchCV is used to to find optimal number of neurons in the layer.
  - Dense Layer is added with ReLu activation. GridSearchCV is used to to find optimal number of neurons in the layer.
  - Dropout Layer with a dropout rate of 0.5.

Adam optimizer was used as solver for all models with the default learning rate at 0.001. GridSearchCV was performed on all models to find the model with highest accuracy and to find the optimal number of neurons in the dense layers for VGG16 and ResNet50. The GridSearchCV was run at 5-folds and 15 epochs. The result of the GridSearchCV are shown in Table 2, and the model with best accuracy was the EfficientNet B0 model with an accuracy of 77%.

Model	Hyper-Parameters	Accuracy
EfficientNetB0	No extra layers to tune	0.7722437500953674
VGG16	Optimal number of nodes in dense layer: 512	0.6541430115699768
ResNet50	Optimal number of nodes in dense layer 1: 256 Optimal number of nodes in dense layer 2: 256	0.679871529340744

Table 2: Results from 5 fold-cross validation and hyper-parameter tuning using GridSearchCV.

4 EXPERIMENTAL SETUP

Environment used for this setup is Jupyter notebook with tensorflow backend and using a nVidia RTX 2070 GPU. The data has been reshaped to have a shape of (5712, 224, 224, 3) for the training set, (1271, 224, 224, 3) for the training set and (40, 224, 224, 3) for the predict dataset. All models used in this paper requires 3 input channels according to the keras documentation[5–7]. The labels of the data, glioma, meningioma, notumor and pituitary need to be encoded into one-hot representation. This is done with the sklearn function to_categorical. This is done so we are able to input the target variables into the CNN models.

- glioma - one-hot representation: [1. 0. 0. 0]
- meningioma - one-hot representation: [0. 1. 0. 0]
- notumor - one-hot representation: [0. 0. 1. 0]
- pituitary - one-hot representation: [0. 0. 0. 1]

Using the results from the GridSearchCV we can now build our models. Table 3, 4 and 5 shows the summary of each model which shows the layers and total number of parameters for each model.

Layer	Output Shape	Number of parameters
EfficientNetB0	7, 7, 1280	4,049,571
Global Average Pooling 2D	1280	0
dropout_47 (Dropout)	1280	0
dense_141 (Dense)	4	5124
Total Params:	4,054,695	
Trainable Params:	4,012,672	
Non-Trainable Params:	42,023	

Table 3: Model summary showing total number of Parameters and layer types used in EfficientNetB0 model.

Layer	Output Shape	Number of parameters
VGG16	7, 7, 512	14,714,688
Flatten_47 (Flatten)	512	0
dense_142 (Dense)	512	262,656
dropout_48 (Dropout)	512	0
dense_143 (Dense)	4	2052
Total Params:	14,979,396	
Trainable Params:	264,708	
Non-Trainable Params:	14,714,688	

Table 4: Model summary showing total number of Parameters and layer types used in VGG16 model.

Layer	Output Shape	Number of parameters
ResNet50	7, 7, 2048	23,587,720
Flatten_46 (Flatten)	2048	0
dense_138 (Dense)	256	524,544
dense_139 (Dense)	256	65,792
dropout_46 (Dropout)	256	0
dense_143 (Dense)	4	1028
Total Params:	24,179,076	
Trainable Params:	24,125,956	
Non-Trainable Params:	53,120	

Table 5: Model summary showing total number of Parameters and layer types used in ResNet50 model.

4.1 Train and Evaluate models

To train our models we need to set a number of parameters like number of epochs, validation split, batch size. We also need to initiate the transfer learning by instantiating the model with weights from the ImageNet database. Once this is done the weights from ImageNet will be downloaded and ready to use. The include top argument is set to be false, this means that we won't include the top layers of the network[6]. The top layer is already trained on the ImageNet dataset and contains the weight form that training. The parameters set for each model for training are validation split set to 0.1 so 90% of the data is used for training and 10% for validation. Dropout rate has set to 0.5 to reduce feature dimension and minimize overfitting[12]. Batch size is set to default, which is 32. Output layers have SoftMax activation function which is used for multi-classification. To make sure the networks converge when training epochs has been set to 50 for this classification problem. We evaluate the model with the test set and using the model predict function. We create a confusion matrix and classification report to evaluate the models performance on the test set.

5 DISCUSSION OF RESULTS

Table 6 shows the accuracy and validation accuracy after training on 50 epochs. EfficientNet B0 validation accuracy is much greater than on VGG16 and ResNet50.

Model Training Accuracy		
Model	Accuracy	Validation Accuracy
EfficientNet B0	99%	95%
VGG16	92%	86%
ResNet50	99%	26%

Table 6: Training accuracy and validation accuracy of each model after 50 epochs.

After training the models we need to evaluate each model. To do this we try to predict the brain tumor classes of the test set X_test. We can create a confusion matrix for each model to visualize the models classifications and misclassifications. The confusion matrix shows the true class on the Y-axis and the predicted class on the X-axis.

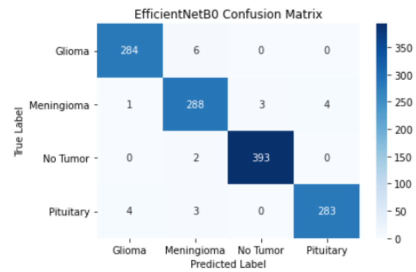

Figure 3: EfficientNet B0 confusion matrix showing the model classifications and misclassifications

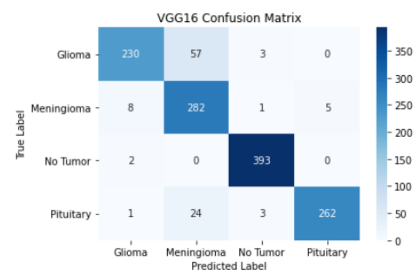

Figure 4: ResNet50 confusion matrix showing the model classifications and misclassifications

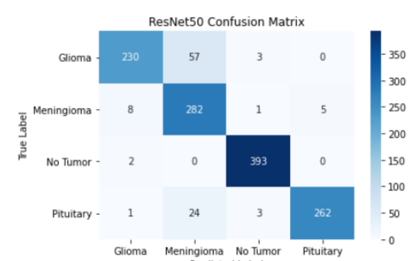

Figure 5: ResNet50 confusion matrix showing the model classifications and misclassifications

Figure 3, 4 and 5 show the confusion matrix of each model. We can see the the EfficientNet model has far less misclassifications than VGG16 and ResNet50. However, all models has misclassified a few images as no tumors, which is not good. We must take a look at the misclassified images to see if there is any mislabeling or corruption within our data or to find another reason for the misclassifications. Figure 6 shows a few samples of the misclassified images done by the most accurate model, EfficientNet.

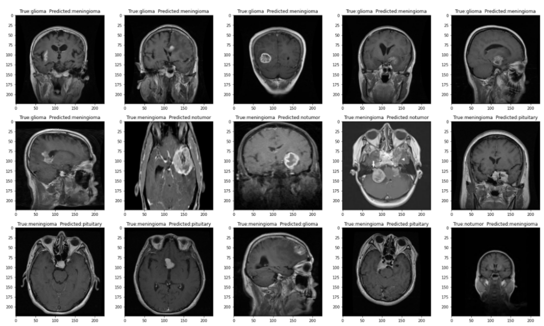

Figure 6: Image samples of misclassified tumors by the EfficientNetB0 model

Looking at the three images where the model predicted no tumor it is apparent that the model has made a mistake. The images are not mislabeled or corrupted. One of the pictures even has an arrow pointing to the tumor. It may be that the model is confused by the color of the tumor and think the tumor is part of the skull instead. Looking at the other misclassifications it can be that this classification model is not able to output the location of the tumors. Being able to output the localization is desired in when processing biomedical images[11].
Table 7 shows the classification report of all three models. Where the most accurate model EfficientNet B0 got an accuracy of 98% when evaluated on the test set. It got a good f1-score of 0.99 on the class no tumor as well, meaning it has a very high accuracy to detect if a tumors is present or not.

Classification Report					
Model	Label	Precision	Recall	f1-score	Accuracy
EfficientNetB0	Glioma	0.98	0.98	0.98	98%
	Meningioma	0.96	0.97	0.97	
	No Tumor	0.99	0.99	0.99	
	Pituitary	0.99	0.98	0.98	
VGG16	Glioma	0.95	0.79	0.87	92%
	Meningioma	0.78	0.95	0.86	
	No Tumor	0.98	0.99	0.99	
	Pituitary	0.98	0.90	0.94	
ResNet50	Glioma	0.95	0.85	0.90	80%
	Meningioma	0.69	0.97	0.81	
	No Tumor	0.78	0.99	0.87	
	Pituitary	1.00	0.32	0.48	

Table 7: Classification report of all models after evaluating each model on the test set. Showing precision, recall, f1-score and accuracy of the models.

We still have the predict dataset to analyze. After we have compared the three models we can now use the one with highest accuracy to try and predict the class labels of the predict dataset. The EfficientNet model predictions can be seen in table 8. We don't have the class labels for this dataset available for evaluation of the prediction.

Picture Number	Model Predictions of predict dataset
1	Glioma
2	Glioma
3	Meningioma
4	Meningioma
5	No Tumor
6	Meningioma
7	Meningioma
8	Meningioma
9	Meningioma
10	Meningioma
11	Meningioma
12	Glioma
13	Meningioma
14	No Tumor
15	No Tumor
16	No Tumor
17	No Tumor
18	No Tumor
19	No Tumor
20	No Tumor
21	No Tumor
22	No Tumor
23	Glioma
24	No Tumor
25	Pituitary
26	Pituitary
27	Meningioma
28	Pituitary
29	Pituitary
30	Pituitary
31	Pituitary
32	Pituitary
33	Pituitary
34	Glioma
35	Pituitary
36	Glioma
37	Glioma
38	Glioma
39	Glioma
40	Glioma

Table 8: Model prediction of classes on the images from the predict dataset.

6 CONCLUSION

This paper compared three pretrained Convolutional Neural Networks EfficientNet B0, VGG16 and ResNet50 for classifying brain tumors. The EfficientNet B0 model achived a promising result of 98% accuracy while VGG16 and ResNet50 achived an accuracy of 92% and 80% respectively. Even when using pretrained models, tuning hyper-parameters and training the models required a fair amount of computational power but is more effective than training the model from scratch. The EfficientNet model has a total of 23 misclassification on this dataset which may be caused by the fact that the classifier are not able to output localization of the tumors. After comparing the three models the one with the highest accuracy (EfficientNet) was used to try and predict the class labels of the predict dataset.

REFERENCES

[1] Karina Biserova, Arvids Jakovlevs, Romans Uljanovs, and Ilze Strumfa. 2021. Cancer Stem Cells: Significance in Origin, Pathogenesis and Treatment of Glioblastoma. *Cells* 10, 3 (2021), 1–20. https://doi.org/10.3390/cells10030621

[2] E. A.S. El-Dahshan, Heba M. Mohsen, Kenneth Revett, and Abdel Badeeh M. Salem. 2014. Computer-aided diagnosis of human brain tumor through MRI: A survey and a new algorithm. *Expert Systems with Applications* 41, 11 (2014), 5526–5545. https://doi.org/10.1016/j.eswa.2014.01.021

[3] Yixing Fu. 2020. Image classification via fine-tuning with EfficientNet. https://keras.io/examples/vision/image_classification_efficientnet_fine_tuning/ Last checked on Oct 05, 2021.

[4] ImageNet. [n. d.]. About ImageNet. https://image-net.org/about.php Last checked on Oct 08, 2021.

[5] Keras. [n. d.]. EfficientNet B0 to B7. https://keras.io/api/applications/efficientnet/#efficientnetb0-function Last checked on Oct 08, 2021.

[6] Keras. [n. d.]. ResNet and ResNetV2. https://keras.io/api/applications/resnet/#resnet50-function Last checked on Oct 05, 2021.

[7] Keras. [n. d.]. VGG16 and VGG19. https://keras.io/api/applications/vgg/#vgg16-function Last checked on Oct 05, 2021.

[8] Alexander Selvikvåg Lundervold and Arvid Lundervold. 2019. An overview of deep learning in medical imaging focusing on MRI. *Zeitschrift fur Medizinische Physik* 29, 2 (2019), 102–127. https://doi.org/10.1016/j.zemedi.2018.11.002 arXiv:1811.10052

[9] Gonçalo Marques, Deevyankar Agarwal, and Isabel de la Torre Díez. 2020. Automated medical diagnosis of COVID-19 through EfficientNet convolutional neural network. *Applied Soft Computing Journal* 96 (2020), 106691. https://doi.org/10.1016/j.asoc.2020.106691

[10] Masoud Nickparvar. 2021. Brain Tumor MRI Dataset. https://www.kaggle.com/masoudnickparvar/brain-tumor-mri-dataset Last checked on Oct 05, 2021.

[11] Olaf Ronneberger Philipp Fischer and Thomas Brox. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *IEEE Access* (may 2015). https://doi.org/10.1109/ACCESS.2021.3053408 arXiv:2105.10559

[12] Venkatesan Rajinikanth, Seifedine Kadry, and Yunyoung Nam. 2021. Convolutional-Neural-Network Assisted Segmentation and SVM Classification of Brain Tumor in Clinical MRI Slices. *Information Technology and Control* 50, 2 (2021), 342–356. https://doi.org/10.5755/j01.itc.50.2.28087

[13] Gopal S. Tandel, Ashish Tiwari, and O. G. Kakde. 2021. Performance optimisation of deep learning models using majority voting algorithm for brain tumour classification. *Computers in Biology and Medicine* 135, February (2021), 104564. https://doi.org/10.1016/j.compbiomed.2021.104564

[14] Y. Le Cun B. Boser J. S. Denker D. Henderson R. E. Howard W. Hubbard and L. D. Jackel. 1990. Handwritten Digit Recognition with a Back-Propagation Network. *Dermatologic surgery : official publication for American Society for Dermatologic Surgery [et al.]* (jan 1990). https://doi.org/10.1111/dsu.12130